PES University Electronic City Campus

Hosur Rd, Konappana Agrahara, Electronic City, Bengaluru, Karnataka 560100

A project Report on

MOVIE TICKET BOOKING SYSTEM

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

UE22CS351A – DBMS Project

Submitted by:

Parvathi Prakash	**PES2UG22CS384**
Raashi Bafna	**PES2UG22CS422**

Under the guidance of

Dr. Mannar Mannan

Associate Professor

PES University

AUG - DEC 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

CERTIFICATE

This is to certify that the mini project entitled

Movie Ticket Booking System

is a bonafide work carried out by

Parvathi Prakash	**PES2UG22CS384**
Raashi Bafna	**PES2UG22CS422**

In partial fulfilment for the completion of fifth semester DBMS Project (UE22CS351A) in the Program of Study - Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period AUG. 2024 – DEC. 2024. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The project has been approved as it satisfies the 5th semester academic requirements in respect of project work.

Signature
Dr. Mannar Mannan
Associate Professor

DECLARATION

We hereby declare that the DBMS Project entitled **University Fest Management System** has been carried out by us under the guidance of **Dr. Mannar Mannan, Associate Professor** and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester AUG – DEC 2024.

Parvathi Prakash	**PES2UG22CS384**
Raashi Bafna	**PES2UG22CS422**

ABSTRACT

- **PROBLEM STATEMENT**:

  In the contemporary entertainment landscape, efficient and user-friendly ticket booking systems are crucial for managing cinema operations and enhancing the customer experience. The current approaches to movie ticket booking often lead to inefficiencies, errors, and suboptimal customer service. There is a need for a comprehensive, automated system that streamlines the booking process, manages inventory effectively, and provides a seamless user experience.

  The online movie ticket booking system is a web-based application designed to streamline the ticket reservation process, offering users a convenient and efficient way to book movie tickets. Built using Python, Streamlit, and MySQL, this system serves as an intuitive platform for customers to view movie schedules, select seats, and purchase tickets online. It also provides theater administrators with functionalities to manage movie listings, schedules, and seat availability.

  This project aims to automate and enhance the traditional ticket booking process by implementing a centralized database that securely manages customer data, booking history, and movie details. Key features include user registration, login, movie selection, seat availability checking, and ticket booking confirmation. Additionally, the system incorporates backend procedures, functions, and triggers to maintain data integrity and streamline operations.

  By simplifying the ticketing process and ensuring seamless interaction between users and administrators, this online booking system offers a modern solution to movie theaters, reducing manual work and improving user experience.

TABLE OF CONTENTS

1. INTRODUCTION

Objective: To design and implement a robust, user-friendly online movie ticket booking system that enables customers to browse available movies, select showtimes, book tickets, and make payments online. The system should also support cinema administrators in managing movie schedules, available seats,etc.

- **SOLUTION**:
  The proposed solution is a comprehensive online movie ticket booking system designed to facilitate an efficient and user-friendly ticket purchasing process for cinema-goers. The system integrates several key components:

  1. **Admin Interface:**
     - *Movie and Showtime Management*: Admins can add, update, or remove movies and schedule showtimes.
  2. **Customer Interface**:
     - *Movie Discovery*: Users can browse and search for movies based on various criteria such as genre, rating, and showtimes.
     - *Seat Selection*: Allows users to view available seats in a graphical layout and select their preferred seats.
     - *Secure Payment*: Facilitates secure online payments and provides electronic tickets.
     - *Booking Management*: Users can view, modify, or cancel their bookings and receive notifications.
     - *Ratings and reviews*: Allows users to read and submit reviews for movies they have watched. Includes a rating system (e.g., 1-5 stars) and text reviews.
     - *Showtime Schedules*: Users can view available showtimes for selected movies.
     - *Profile Management*: Users can create and manage profiles, including personal information, payment methods, and preferences.
  3. **Backend System:**
     - *Database Management*: A relational database to manage movie data, user bookings, and transactions.
     - *Integration*: Includes integration with payment gateways for secure transactions.

Scalability: Designed to handle high volumes of traffic and bookings, especially during peak times.

2. PROBLEM DEFINITION WITH USER REQUIREMENT SPECIFICATIONS

Problem Definition

The movie theater industry continues to face challenges with traditional in-person ticket booking methods. These methods often result in long wait times, limited show availability by the time a customer reaches the counter, and inefficiencies in managing seat allocations. Additionally, many customers prefer the convenience of online booking, enabling them to plan their movie experience in advance and avoid sold-out screenings. This shift has created a need for an efficient online system that allows users to reserve tickets anytime, anywhere.

The current lack of an automated, centralized system makes it difficult for theaters to manage real-time data on show timings, seating availability, and booking records. Furthermore, without a digital solution, tracking customer data, managing movie schedules, and analyzing booking trends for strategic planning become cumbersome.

The objective of this project is to develop a comprehensive online movie ticket booking system that resolves these issues by providing customers with a convenient, user-friendly platform for reserving seats and theaters with an efficient backend system for managing bookings, show schedules, and seating. This solution will reduce administrative workload, minimize booking errors, and improve customer satisfaction by offering a streamlined booking experience.

User Requirement Specifications

The online movie ticket booking system involves two main user roles: **Customer** and **Admin**, each with specific requirements as follows:

1. User Roles and Access

- **Customer**:
  - Can register and create a profile.
  - Has access to browse movies, view showtimes, select seats, book tickets, and review their booking history.
  - Receives booking confirmations and digital tickets.
- **Admin**:
  - Has full access to manage movie and show information, including adding, updating, and deleting movie details.
  - Manages show timings, seating configurations, and booking records.

2. Functional Requirements

- **User Registration and Login**

  - **User Registration**: Customers register with their email, username, and password, creating an account in the system.
  - **Authentication**: Secure login for both customers and admins with role-based access, ensuring proper access control.
- **Movie and Show Listings**
  - **Movie Details**: Display detailed information on movies, including title, genre, duration, rating, and synopsis.
  - **Showtimes**: Display available screening times for each movie, along with the assigned screen and seat availability.
- **Show Details and Seat Selection**
  - **Seat Availability**: Show seat map for each screening, indicating available and occupied seats in real time.
  - **Seat Selection**: Customers can select specific seats and booking.
- **Ticket Booking and Payment**
  - **Booking Process**: Provide a straightforward process for customers to select a movie, showtime, and seat, and complete booking with a payment confirmation.
  - **Payment Confirmation**: Ensure secure handling of payment.
- **Booking History and Management**
  - **Booking History**: Customers can view their past and upcoming bookings, with details of movie title, showtime, and seat information.
- **Admin Management Panel**
  - **Movie and Show Management**: Admins can add, update, and delete movies, adjust show timings, and manage screening assignments.
  - **Seating Management**: Admins can configure seat maps for each screen and adjust seating arrangements if needed.

3. Non-Functional Requirements

- **Usability**
  - **Ease of Use**: A clear, user-friendly interface that allows users to browse, book, and manage tickets easily.
- **Reliability**
  - **Data Accuracy**: Provide accurate, up-to-date information on seat availability and show schedules.
  - **System Uptime**: Ensure high availability of the system to accommodate customers booking tickets at any time.
- **Performance**
  - **Response Time**: Optimize for fast load times and responsiveness, especially when displaying seat availability and processing bookings.
  - **Scalability**: Design the system to handle a large number of simultaneous users without performance degradation.
- **Security**
  - **Data Security**: Protect customer information and payment details using secure encryption and ensure compliance with industry standards.
  - **Role-Based Access Control**: Implement strict access control to safeguard admin functionalities and prevent unauthorized access to the backend.
  - **Transaction Security**: Ensure all payments and booking transactions are processed securely, with confirmation messages sent to users.

3. LIST OF SOFTWARES/TOOLS/PROGRAMMING LANGUAGES USED

1. Programming Languages

- **Python**: Main programming language for developing the backend functionality, business logic, and server-side processing.
- **SQL**: For database queries, schema creation, and data manipulation in MySQL.

2. Frameworks and Libraries

- **Streamlit**: A Python framework for building and deploying the web interface, allowing users to interact with the booking system.
- **Pandas**: For data manipulation and analysis, used to handle and process user and booking data if required.
- **MySQL Connector for Python**: Library to connect Python with MySQL, facilitating database operations from the application.

3. Database Management System

- **MySQL**: Relational database management system to store, manage, and query data related to movies, bookings, users, and show schedules.

4. Development Environment and Tools

- **VS Code**: Integrated development environments (IDEs) for writing, testing, and debugging Python code.
- **MySQL Workbench**: For designing and managing database schemas, creating stored procedures, functions, and triggers.

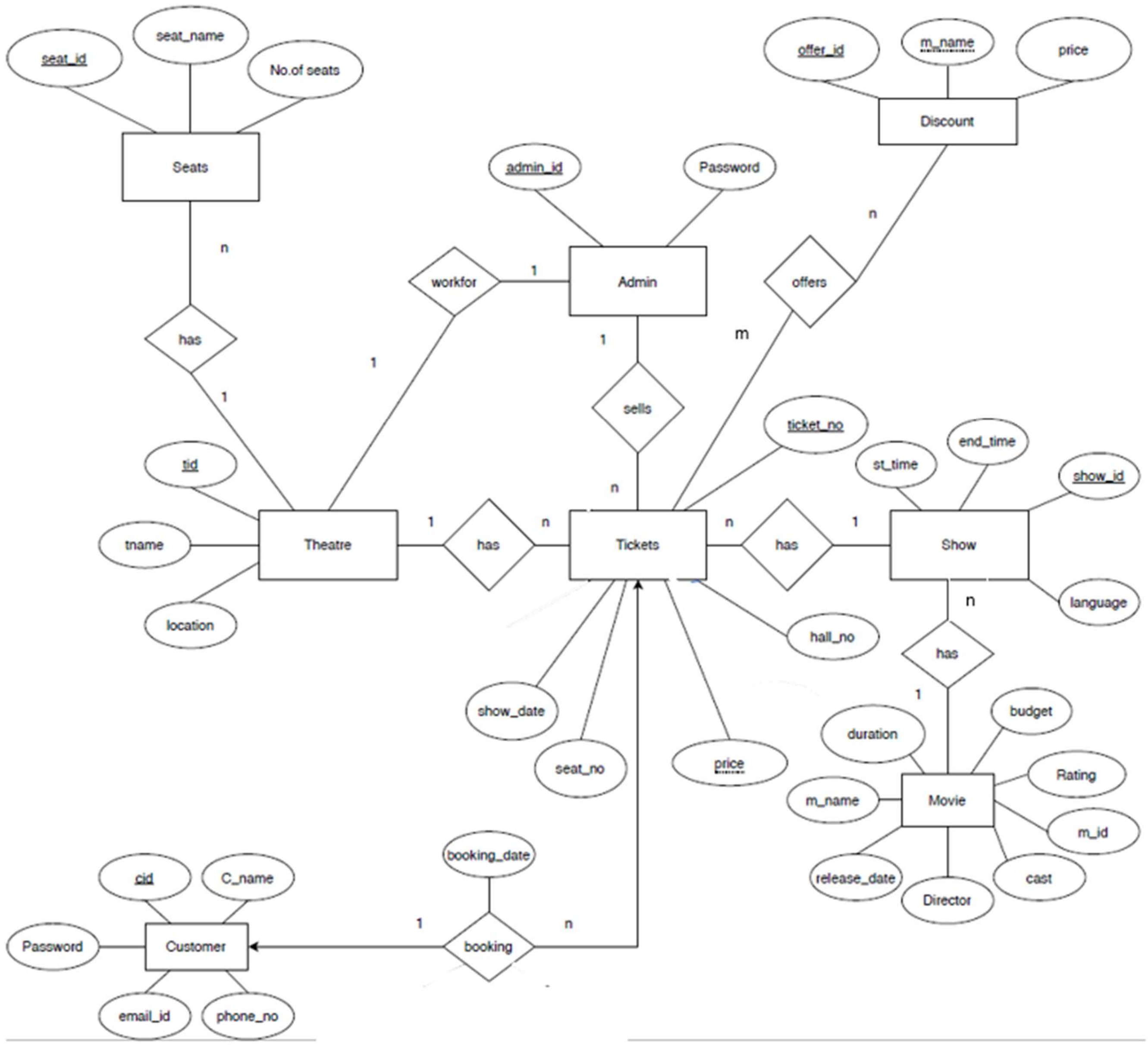
5. Web Hosting and Deployment (Optional)

- **Streamlit Cloud**: For deploying the web application online, making it accessible to end-users over the internet.

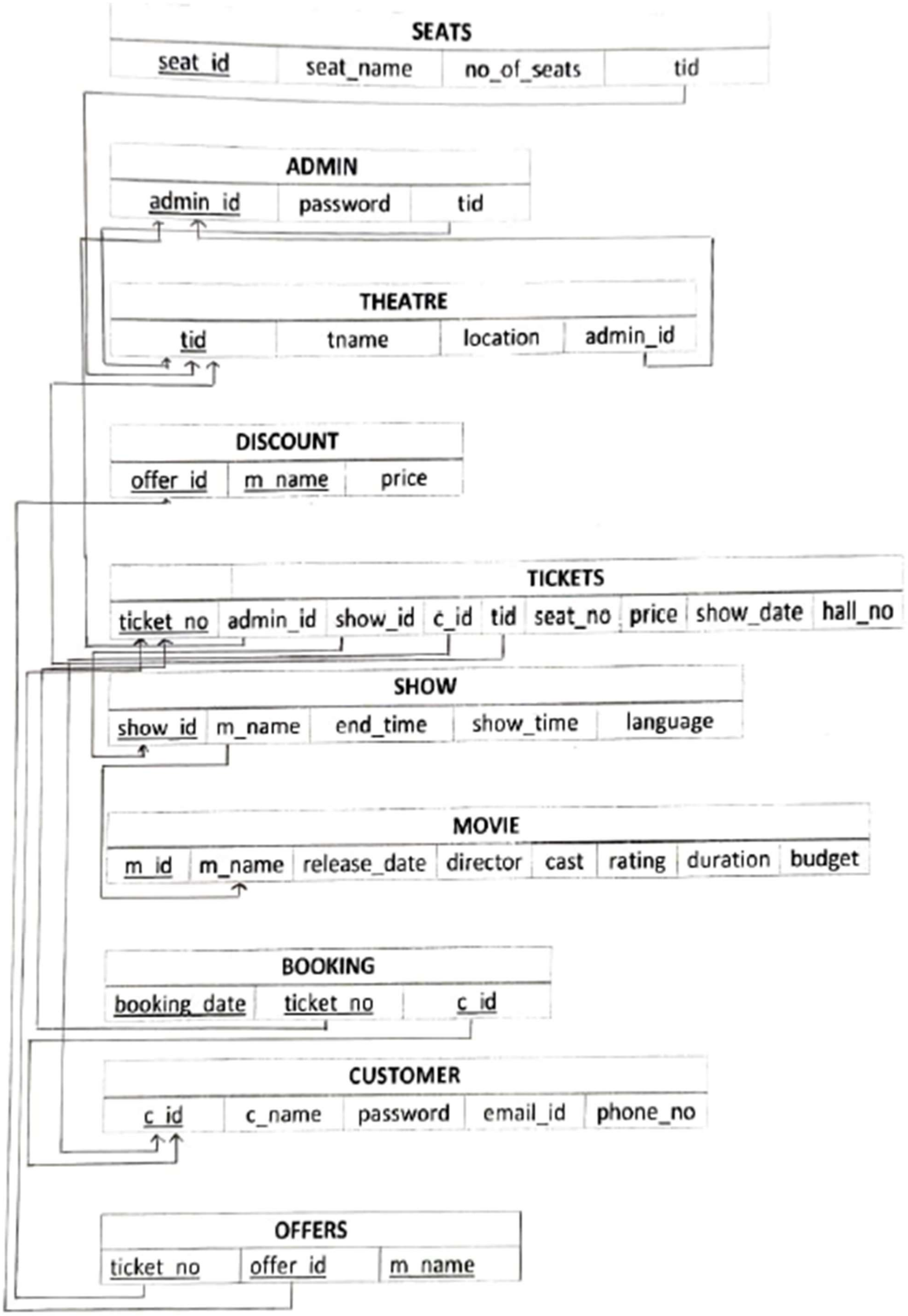
6. Version Control

- **Git**: For version control to track code changes, collaborate with other developers, and manage different project versions.
- **GitHub / GitLab**: Online repository platforms to store, manage, and share code, and to handle collaborative development.

4. ER MODEL


seat_id
seat_name
No.of seats
Seats
n
has
1
offer_id
m_name
price
Discount
n
admin_id
Password
Admin
workfor
1
1
offers
m
1
sells
n
ticket_no
end_time
st_time
show_id
tid
tname
Theatre
location
1
has
n
Tickets
n
has
1
Show
language
n
hall_no
has
1
show_date
seat_no
price
duration
budget
Rating
m_name
Movie
m_id
release_date
Director
cast
booking_date
cid
C_name
Password
Customer
1
booking
n
email_id
phone_no

5. ER TO RELATIONAL MAPPING



6. DDL STATEMENTS

```
CREATE TABLE screens (screen_id INT NOT NULL PRIMARY KEY, class
varchar(10) NOT NULL, no_of_seats INT);
```

```
CREATE TABLE movies ( movie_id INT NOT NULL PRIMARY KEY, poster_link
VARCHAR(255), movie_name VARCHAR(50) NOT NULL, released_year INT,
language VARCHAR(20),
runtime INT, genre VARCHAR(100), imdb_rating DECIMAL(2, 1),overview TEXT,
director VARCHAR(100), star1 VARCHAR(100), star2 VARCHAR(100), show_start
DATE, show_end DATE);
```

```
CREATE TABLE price_list (price_id INT NOT NULL PRIMARY KEY, type INT, day
VARCHAR(10), price INT);
```

```
CREATE TABLE shows (show_id INT NOT NULL PRIMARY KEY, type INT, time
VARCHAR(10), date DATE, screen_id INT, movie_id INT, price_id INT, foreign
key(movie_id) REFERENCES movies(movie_id) ON DELETE CASCADE ON
UPDATE CASCADE, foreign key(screen_id) REFERENCES screens(screen_id) ON
DELETE CASCADE ON UPDATE CASCADE, foreign key(price_id) REFERENCES
price_list(price_id) ON UPDATE CASCADE ON DELETE CASCADE);
```

```
CREATE TABLE booked_tickets (ticket_no INT NOT NULL, seat_no INT NOT
NULL, show_id INT NOT NULL, payment_status VARCHAR(20) DEFAULT
'PENDING', primary key(ticket_no,show_id), foreign key(show_id) REFERENCES
shows(show_id) ON DELETE CASCADE ON UPDATE CASCADE);
```

```
CREATE TABLE user (uname VARCHAR(30), uid INT NOT NULL PRIMARY
KEY, password VARCHAR(255) NOT NULL);
```

```
CREATE TABLE admin (aid INT NOT NULL PRIMARY KEY, aname
VARCHAR(30), password VARCHAR(30));
```

```
CREATE TABLE has_booked (ticket_no INT NOT NULL, uid INT NOT NULL,
primary key(ticket_no,uid), foreign key(ticket_no) REFERENCES
booked_tickets(ticket_no) ON UPDATE CASCADE ON DELETE CASCADE, foreign
key(uid) REFERENCES user(uid) ON UPDATE CASCADE ON DELETE CASCADE
);
```

```
CREATE TABLE manages(aid INT NOT NULL, show_id INT NOT NULL, primary
key(aid,show_id), foreign key(aid) REFERENCES admin(aid) ON UPDATE
CASCADE ON DELETE CASCADE, foreign key(show_id) REFERENCES
shows(show_id) ON UPDATE CASCADE ON DELETE CASCADE);
```

7. DML STATEMENTS (CRUD OPERATION SCREENSHOTS)

```
insert into screens values(1,'platinum',100);
insert into screens values(2,'standard',150);
insert into screens values(3,'gold',120);
insert into screens values(4,'platinum',50);
insert into screens values(5,'gold',100);
```

```
INSERT INTO movies (movie_id, poster_link, movie_name, released_year, language,
runtime, genre, imdb_rating, overview, director, star1, star2, show_start, show_end)
VALUES
(101, 'https://th.bing.com/th/id/OIP.d7ndMFO3WSnU5db361fvSQHaK-
?w=115&h=180&c=7&r=0&o=5&dpr=1.3&pid=1.7', 'Bhool Bhulaiyaa 3', 2024,
'Hindi', 158, 'horror, comedy', 5.9, 'Ruhaan, a fraudster posing as an exorcist, takes on a
lucrative case at a haunted castle, unraveling a sinister plot involving mischievous
priests, culminating in a hilarious yet thrilling ride filled with unexpected twists and
scares.', 'Anees Bazmee', 'Kartik Aryan', 'Vidya Balan', '2024-11-01', '2024-12-31');

INSERT INTO movies (movie_id, poster_link, movie_name, released_year, language,
runtime, genre, imdb_rating, overview, director, star1, star2, show_start, show_end)
VALUES
(102,
'https://th.bing.com/th/id/OIP.c2Acbosdb65pXSv4QTp7jAHaLk?w=115&h=180&c=7&
r=0&o=5&dpr=1.3&pid=1.7', 'Amaran', 2024, 'Tamil', 169, 'action, drama', 8.6, 'The life
of Major Mukund Varadarajan and is set against the backdrop of the Qazipathri
Operation in Shopian, Kashmir, which took place back in 2014.', 'Rajkumar Periasamy',
'Sivakartikeyan', 'Sai Pallavi', '2024-10-31', '2024-11-30');

INSERT INTO movies (movie_id, poster_link, movie_name, released_year, language,
runtime, genre, imdb_rating, overview, director, star1, star2, show_start, show_end)
VALUES
(103, 'https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcTx0W0sS1kHEvF4hGQURe6awxrbtX-
XEe55UA&s', 'Singham Again', 2024, 'Hindi', 144, 'action, drama', 6.8, 'A new chase is
coming - with reference to the epic Ramayana, Singham and his team face an ambiguous
villain in order to save his wife.', 'Rohit Shetty', 'Ajay Devgn', 'Akshay Kumar', '2024-
11-01', '2024-12-10');

INSERT INTO movies (movie_id, poster_link, movie_name, released_year, language,
runtime, genre, imdb_rating, overview, director, star1, star2, show_start, show_end)
```

```
VALUES
(104,
'https://th.bing.com/th/id/OIP.uXqZ7t6xWYixmD7A8GL3gAHaJQ?w=128&h=180&c=
7&r=0&o=5&dpr=1.3&pid=1.7', 'Venom: The last dance', 2024, 'English', 110, 'action,
adventure', 6.2, 'Eddie and Venom, on the run, face pursuit from both worlds. As
circumstances tighten, they re compelled to make a heart-wrenching choice that could
mark the end of their symbiotic partnership.', 'Kelly Marcel', 'Tom Hardy', 'Chiwetel
Ejiofor', '2024-10-24', '2024-12-15');

INSERT INTO movies (movie_id, poster_link, movie_name, released_year, language,
runtime, genre, imdb_rating, overview, director, star1, star2, show_start, show_end)
VALUES
(105,
'https://th.bing.com/th/id/OIP.vJMeZv7_1FlGYtislD6yVwHaLH?w=118&h=180&c=7
&r=0&o=5&dpr=1.3&pid=1.7', 'Vettaiyan - The Hunter', 2024, 'Tamil', 163, 'action,
drama', 7.9, 'A sought-after supercop gets caught in a series of unexpected events when
he guns down a criminal in a murder case.', 'T.J. Gnanavel', 'Rajinikanth', 'Fahadh
Faasil', '2024-10-10', '2024-12-20');

INSERT INTO movies (movie_id, poster_link, movie_name, released_year, language,
runtime, genre, imdb_rating, overview, director, star1, star2, show_start, show_end)
VALUES
(106,
'https://th.bing.com/th/id/OIP.UV5FDSDrcs8x8TWEO84KtQHaK9?w=115&h=180&c=
7&r=0&o=5&dpr=1.3&pid=1.7', 'Rebel Moon - Part Two: The Scargiver', 2024,
'English', 122, 'action, adventure', 5.6, 'Kora and surviving warriors prepare to defend
Veldt, their new home, alongside its people against the Realm.', 'Zack Snyder', 'Sofia
Boutella', 'Djimon Hounsou', '2024-11-05', '2024-11-15');

INSERT INTO movies (movie_id, poster_link, movie_name, released_year, language,
runtime, genre, imdb_rating, overview, director, star1, star2, show_start, show_end)
VALUES
(107, 'https://th.bing.com/th/id/OIP.9HPL3jOwZOCFS-
6cF_4VXgHaLH?w=115&h=180&c=7&r=0&o=5&dpr=1.3&pid=1.7', 'Manjummel
Boys', 2024, 'Malayalam', 135, 'adventure, drama', 8.2, 'A group of friends get into a
daring rescue mission to save their friend from Guna Caves, a perilously deep pit from
where nobody has ever been brought back.', 'Chidambaram', 'Soubin Shahir', 'Sreenath
Bhasi', '2024-11-06', '2024-11-16');

INSERT INTO movies (movie_id, poster_link, movie_name, released_year, language,
runtime, genre, imdb_rating, overview, director, star1, star2, show_start, show_end)
VALUES
```

```
(108, 'https://th.bing.com/th/id/OIP.qLV2cMRF6Of-
5pnjqToQUQHaJT?w=208&h=261&c=7&r=0&o=5&dpr=1.3&pid=1.7', 'Inside Out 2',
2024, 'English', 96, 'adventure, drama', 7.6, 'A sequel that features Riley entering puberty
and experiencing brand new, more complex emotions as a result.', 'Kelsey Mann', 'Amy
Poehler', 'Maya Hawke', '2024-11-07', '2024-11-19');

INSERT INTO movies (movie_id, poster_link, movie_name, released_year, language,
runtime, genre, imdb_rating, overview, director, star1, star2, show_start, show_end)
VALUES
(109,
'https://th.bing.com/th/id/OIP.UrjJnOLTonimB4bkvndAMwAAAA?w=205&h=304&c=
7&r=0&o=5&dpr=1.3&pid=1.7', 'Ghostbusters: Frozen Empire', 2024, 'English', 135,
'adventure, comedy', 6.1, 'When the discovery of an ancient artifact unleashes an evil
force, Ghostbusters new and old must join forces to protect their home and save the
world from a second ice age.', 'Gil Kenan', 'Paul Rudd', 'Carrie Coon', '2024-11-10',
'2024-11-20');

INSERT INTO movies (movie_id, poster_link, movie_name, released_year, language,
runtime, genre, imdb_rating, overview, director, star1, star2, show_start, show_end)
VALUES
(110,
'https://th.bing.com/th/id/OIP.5cWtBlgKq9O01FLR810uLQHaOG?w=115&h=180&c=
7&r=0&o=5&dpr=1.3&pid=1.7', 'Stree 2', 2024, 'Hindi', 147, 'horror, comedy', 7.1,
'After the events of Stree, the town of Chanderi is being haunted again. This time,
women are mysteriously abducted by a terrifying headless entity.', 'Amar Kaushik',
'Rajkumar Rao', 'Shraddha Kapoor', '2024-11-11', '2024-11-24');


insert into price_list values(1,3,'Friday',500);
insert into price_list values(2,3,'Saturday',500);
insert into price_list values(3,2,'Friday',350);
insert into price_list values(4,2,'Saturday',350);
insert into price_list values(5,1,'Sunday',450);

insert into shows values(001,3,'12:00:00','2024-11-17',1,101,1);
insert into shows values(002,2,'11:30:00','2024-11-17',3,103,2);
insert into shows values(003,3,'12:15:00','2024-11-18',1,101,1);
insert into shows values(004,2,'15:30:00','2024-11-19',2,102,3);
insert into shows values(005,1,'21:45:00','2024-11-20',2,102,2);

INSERT INTO booked_tickets (ticket_no, seat_no, show_id) VALUES (170, 15, 1);
INSERT INTO booked_tickets (ticket_no, seat_no, show_id) VALUES (64, 85, 2);
```

INSERT INTO booked_tickets (ticket_no, seat_no, show_id) VALUES (10, 90, 2);
INSERT INTO booked_tickets (ticket_no, seat_no, show_id) VALUES (22, 54, 3);
INSERT INTO booked_tickets (ticket_no, seat_no, show_id) VALUES (35, 100, 1);

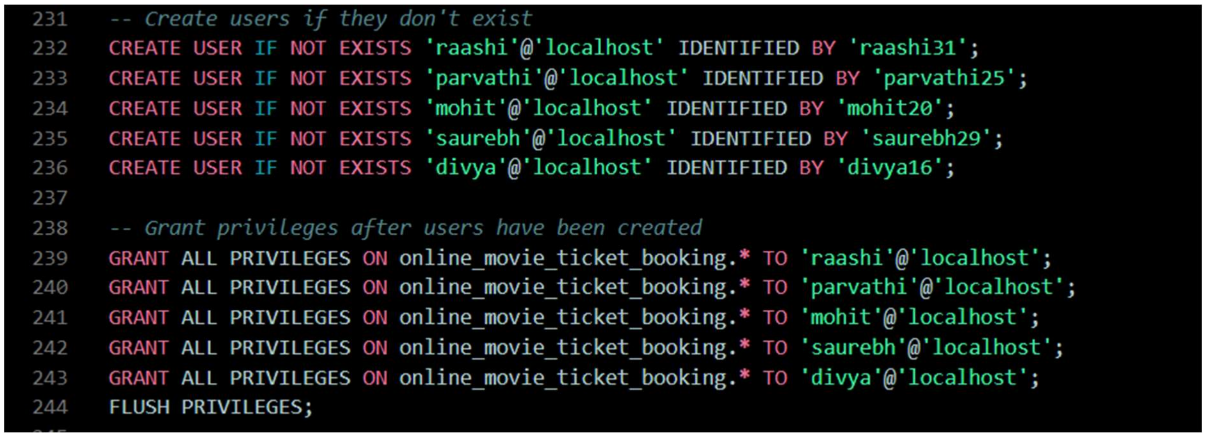
insert into user values('Tarun',700,'tarun700');
insert into user values('Neeraja',018,'neeraja018');
insert into user values('Rachana',007,'rachana007');
insert into user values('Yash',010,'yash010');
insert into user values('Ronak',065,'ronak065');
insert into user values('pepper',01,'pepper01');

insert into admin values(001,'raashi','raashi31');
insert into admin values(002,'parvathi','parvathi25');
insert into admin values(003,'mohit','mohit20');
insert into admin values(004,'saurebh','saurebh29');
insert into admin values(005,'divya','divya16');

insert into has_booked values(0170,018);
insert into has_booked values(0064,700);
insert into has_booked values(0010,007);
insert into has_booked values(0035,065);
insert into has_booked values(0022,010);

insert into manages values(001,001);
insert into manages values(001,002);
insert into manages values(002,002);
insert into manages values(003,003);
insert into manages values(004,001);
insert into manages values(005,002);

DCL statements:



```sql
-- Create users if they don't exist
CREATE USER IF NOT EXISTS 'raashi'@'localhost' IDENTIFIED BY 'raashi31';
CREATE USER IF NOT EXISTS 'parvathi'@'localhost' IDENTIFIED BY 'parvathi25';
CREATE USER IF NOT EXISTS 'mohit'@'localhost' IDENTIFIED BY 'mohit20';
CREATE USER IF NOT EXISTS 'saurebh'@'localhost' IDENTIFIED BY 'saurebh29';
CREATE USER IF NOT EXISTS 'divya'@'localhost' IDENTIFIED BY 'divya16';

-- Grant privileges after users have been created
GRANT ALL PRIVILEGES ON online_movie_ticket_booking.* TO 'raashi'@'localhost';
GRANT ALL PRIVILEGES ON online_movie_ticket_booking.* TO 'parvathi'@'localhost';
GRANT ALL PRIVILEGES ON online_movie_ticket_booking.* TO 'mohit'@'localhost';
GRANT ALL PRIVILEGES ON online_movie_ticket_booking.* TO 'saurebh'@'localhost';
GRANT ALL PRIVILEGES ON online_movie_ticket_booking.* TO 'divya'@'localhost';
FLUSH PRIVILEGES;
```

8. QUERIES (JOIN QUERY, AGGREGATE FUNCTION QUERIES AND NESTED QUERY)

```
SELECT genre, movie_name, imdb_rating
FROM movies
WHERE imdb_rating = (
   SELECT MAX(imdb_rating)
   FROM movies AS m2
   WHERE m2.genre = movies.genre
);
```

```
SELECT b.ticket_no, u.uname AS username, m.movie_name, s.time AS showtime,
b.seat_no
FROM booked_tickets b
JOIN has_booked hb ON b.ticket_no = hb.ticket_no
JOIN user u ON hb.uid = u.uid
JOIN shows s ON b.show_id = s.show_id
JOIN movies m ON s.movie_id = m.movie_id;
```

```
SELECT s.movie_id, COUNT(*) AS total_bookings
FROM booked_tickets b
JOIN shows s ON b.show_id = s.show_id
GROUP BY s.movie_id;
```

```
SELECT day, MAX(price) AS max_price, MIN(price) AS min_price FROM price_list
GROUP BY day;
```

9. STORED PROCEDURE, FUNCTIONS AND TRIGGERS

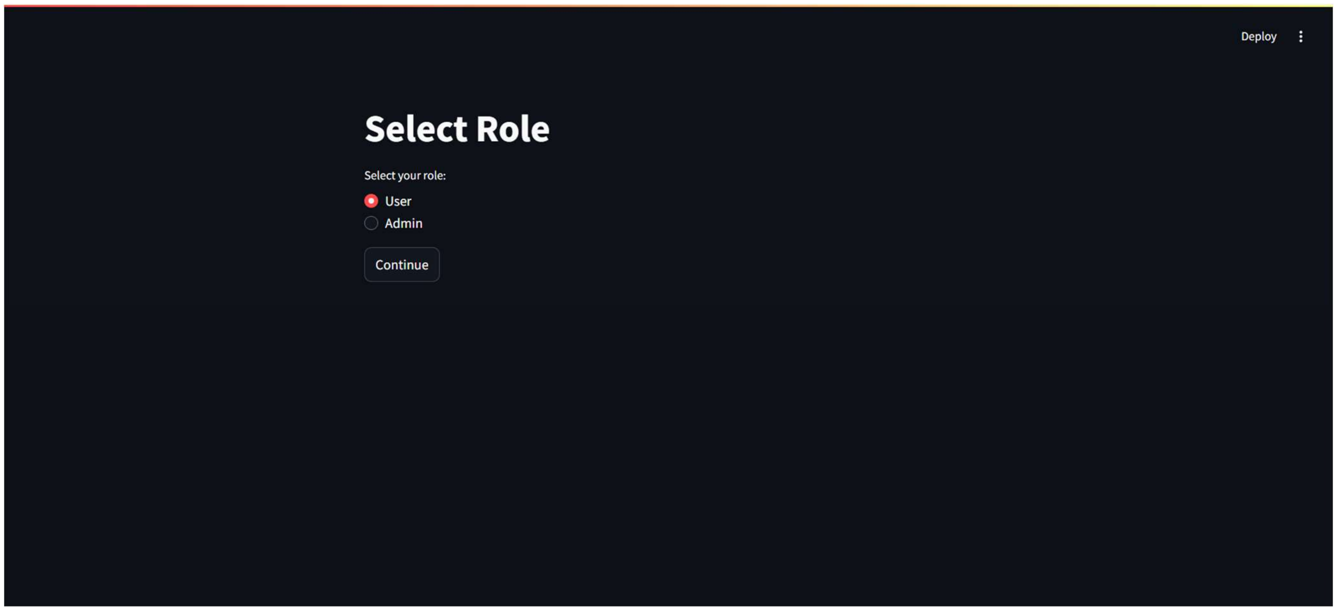
```
DELIMITER //

CREATE FUNCTION no_of_freeseats(showId INT)
RETURNS INT
DETERMINISTIC
READS SQL DATA
BEGIN
   DECLARE total_seats INT;
   DECLARE booked_seats INT;
   DECLARE free_seats INT;

   SELECT s.no_of_seats INTO total_seats
   FROM screens s
   JOIN shows sh ON s.screen_id = sh.screen_id
   WHERE sh.show_id = showId;

   SELECT COUNT(*) INTO booked_seats
   FROM booked_tickets
   WHERE show_id = showId;

   SET free_seats = total_seats - booked_seats;

   RETURN free_seats;
END //
DELIMITER ;


DELIMITER //
CREATE TRIGGER capacity_check BEFORE UPDATE ON screens
FOR EACH ROW
```

```
BEGIN
    IF NEW.no_of_seats < 0 OR NEW.no_of_seats > 300 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Capacity must be between 0 and 300.';
    END IF;
END //
DELIMITER ;

DELIMITER //


CREATE PROCEDURE show_booked_tickets(IN input_show_id INT)
BEGIN
    -- Select booked tickets for the given show_id
    SELECT ticket_no, seat_no
    FROM booked_tickets
    WHERE show_id = input_show_id;
END //

DELIMITER ;
-- Procedure to book a ticket
DELIMITER //
CREATE PROCEDURE BookTicket(
    IN p_uid INT,
    IN p_show_id INT,
    IN p_ticket_no INT,
    IN p_seat_no INT
)
BEGIN
    -- Insert booking details into booked_tickets using provided ticket_no
    INSERT INTO booked_tickets (ticket_no, seat_no, show_id)
    VALUES (p_ticket_no, p_seat_no, p_show_id);
    -- Insert into has_booked to link the user with the ticket
    INSERT INTO has_booked (ticket_no, uid)
    VALUES (p_ticket_no, p_uid);

    SELECT 'Booking confirmed' AS Status;
END //
DELIMITER ;

-- Trigger to insert a notification when a booking is made
DROP TRIGGER IF EXISTS after_booking_insert;
DELIMITER //
CREATE TRIGGER after_booking_insert
AFTER INSERT ON booked_tickets
FOR EACH ROW
BEGIN
    INSERT INTO notifications (uid, message)
    VALUES (
        (SELECT uid FROM has_booked WHERE ticket_no = NEW.ticket_no),
        CONCAT('Booking confirmed for show ID: ', NEW.show_id, ' - Seat: ', NEW.seat_no)
    );
END //
DELIMITER ;

-- Procedure to confirm a booking
DELIMITER //
CREATE PROCEDURE confirm_booking(IN p_ticket_no INT, IN p_show_id INT)
BEGIN
    UPDATE booked_tickets
    SET payment_status = 'CONFIRMED'
    WHERE ticket_no = p_ticket_no AND show_id = p_show_id;
END //
DELIMITER ;
```

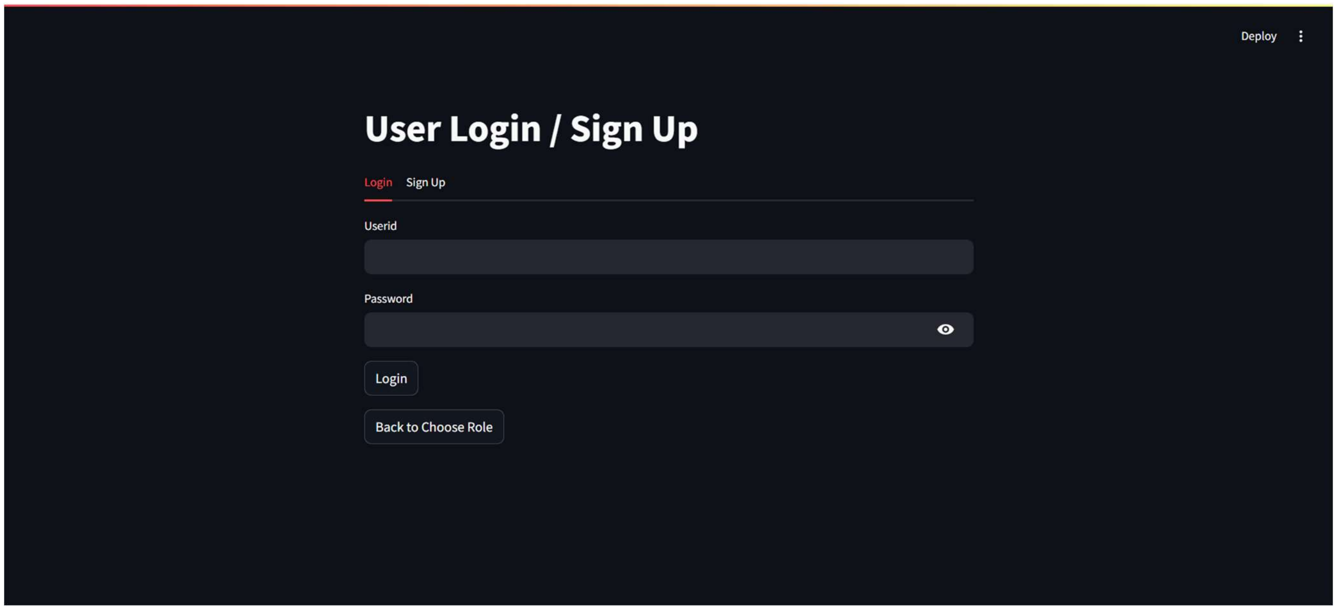
10. FRONT END DEVELOPMENT (FUNCTIONALITIES/FEATURES OF THE APPLICATION)

User interface :

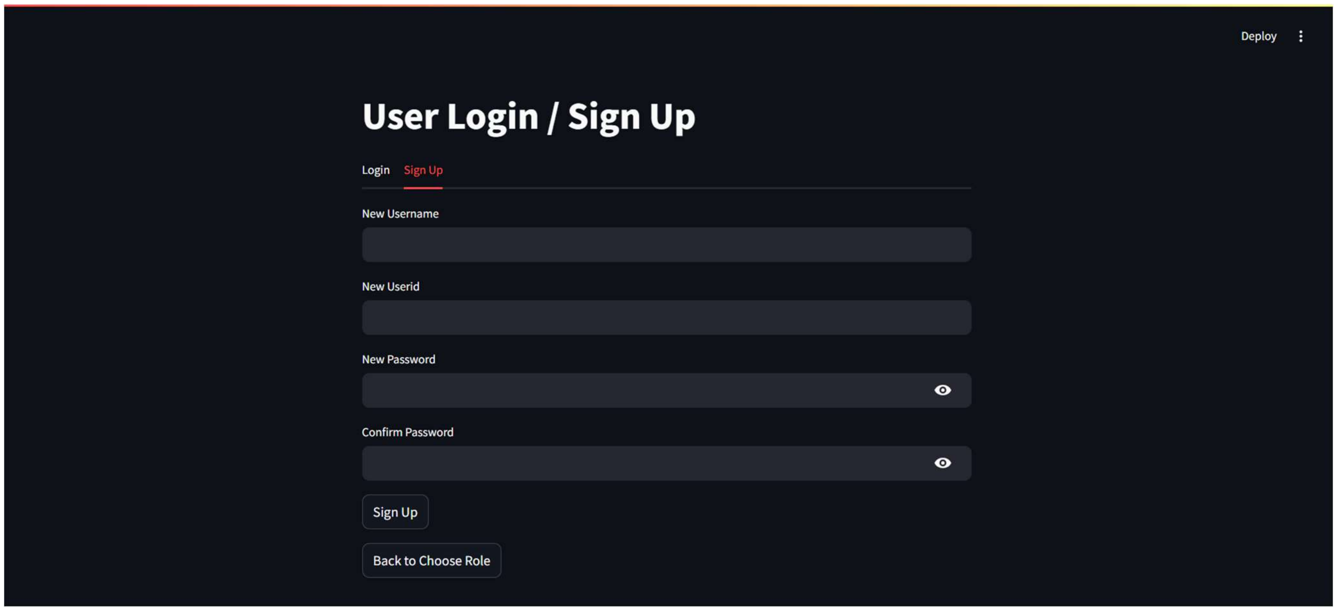
- Choose role :



- User Login :



- User sign up :



Deploy

User Login / Sign Up

Login Sign Up

New Username

New Userid

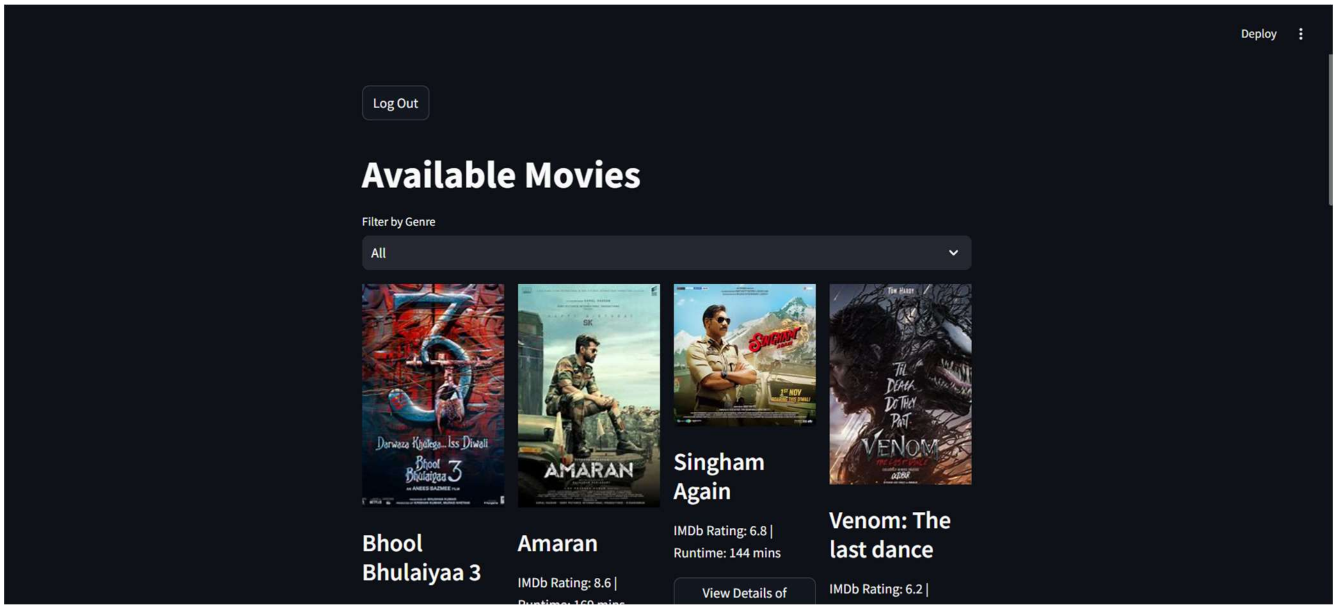
New Password

Confirm Password

Sign Up

Back to Choose Role

- Movie browser :



Deploy

Log Out

Available Movies

Filter by Genre

All

Bhool Bhulaiyaa 3

Amaran

IMDb Rating: 8.6 |

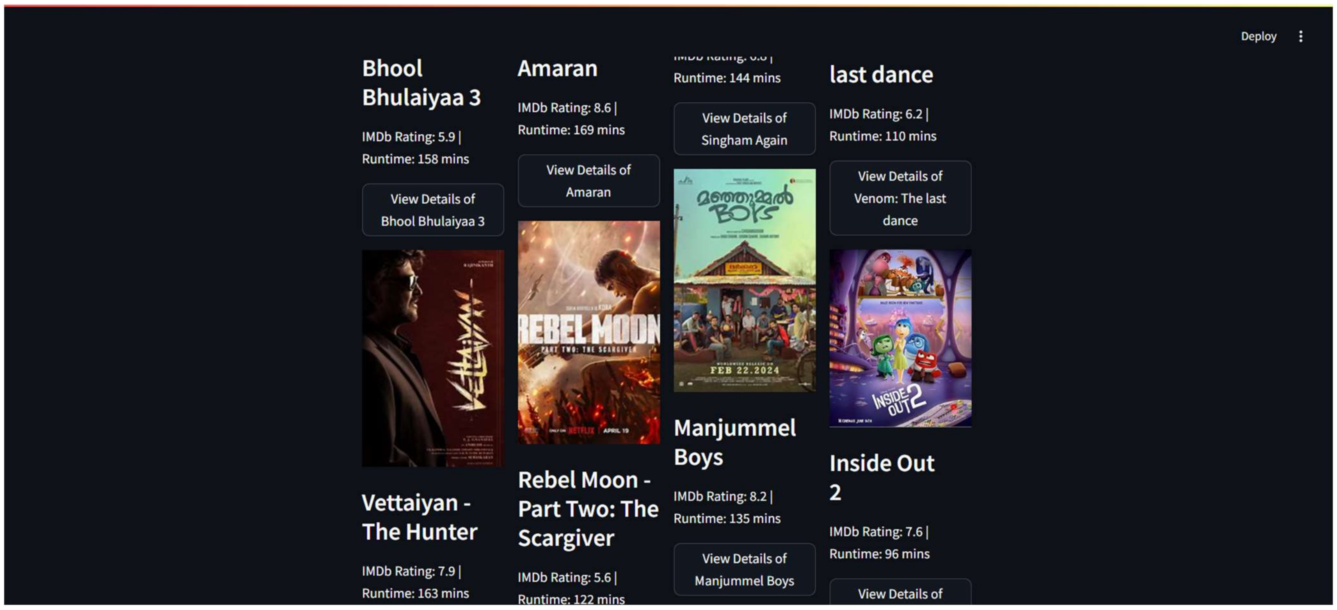
Singham Again

IMDb Rating: 6.8 |
Runtime: 144 mins

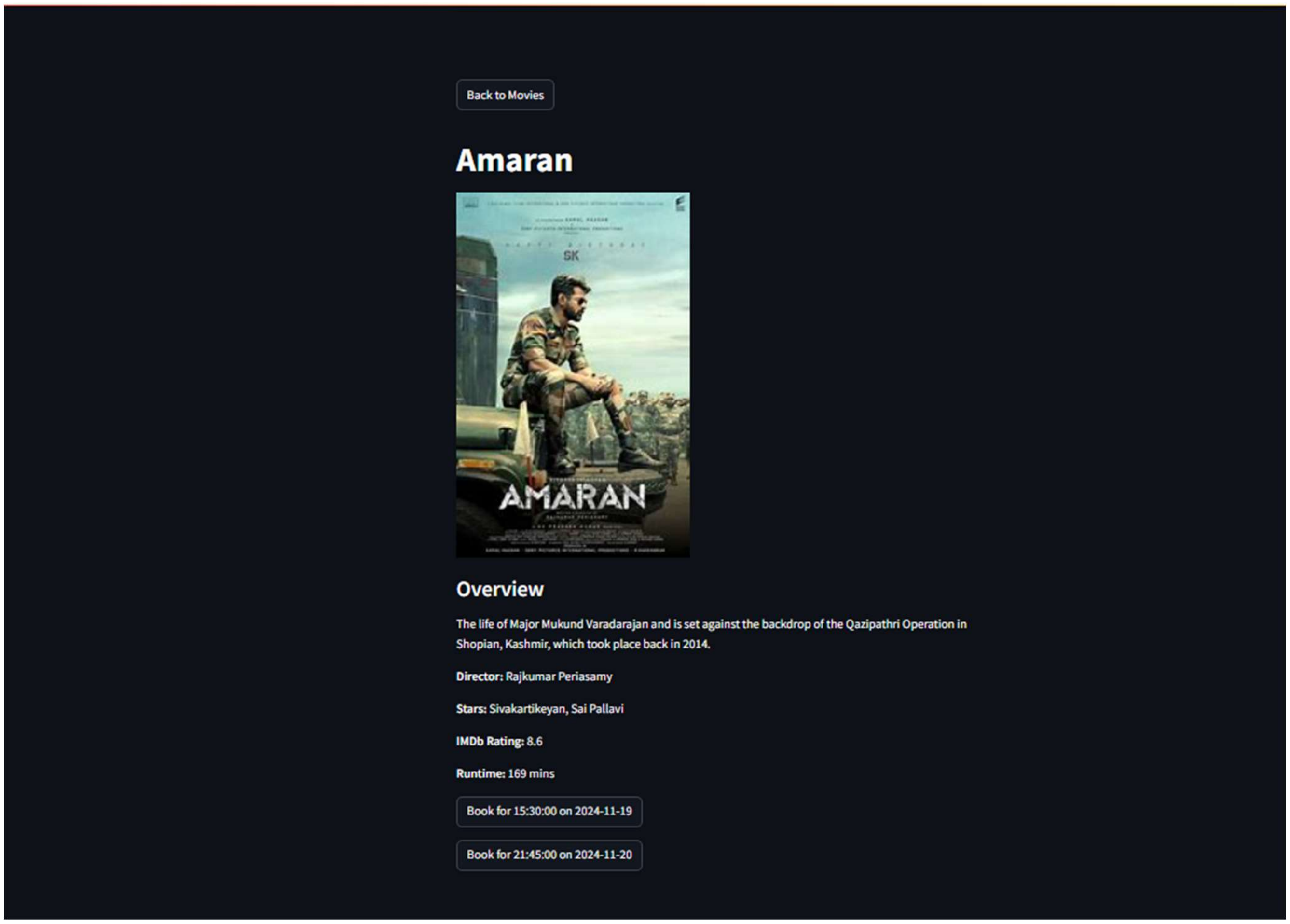
View Details of

Venom: The last dance

IMDb Rating: 6.2 |


Deploy
Bhool Bhulaiyaa 3
IMDb Rating: 5.9 | Runtime: 158 mins
View Details of Bhool Bhulaiyaa 3
Amaran
IMDb Rating: 8.6 | Runtime: 169 mins
View Details of Amaran
Runtime: 144 mins
View Details of Singham Again
last dance
IMDb Rating: 6.2 | Runtime: 110 mins
View Details of Venom: The last dance
Vettaiyan - The Hunter
IMDb Rating: 7.9 | Runtime: 163 mins
Rebel Moon - Part Two: The Scargiver
IMDb Rating: 5.6 | Runtime: 122 mins
Manjummel Boys
IMDb Rating: 8.2 | Runtime: 135 mins
View Details of Manjummel Boys
Inside Out 2
IMDb Rating: 7.6 | Runtime: 96 mins
View Details of

- Movie details :


Back to Movies
Amaran
AMARAN
Book for 15:30:00 on 2024-11-19
Book for 21:45:00 on 2024-11-20

Overview

The life of Major Mukund Varadarajan and is set against the backdrop of the Qazipathri Operation in Shopian, Kashmir, which took place back in 2014.

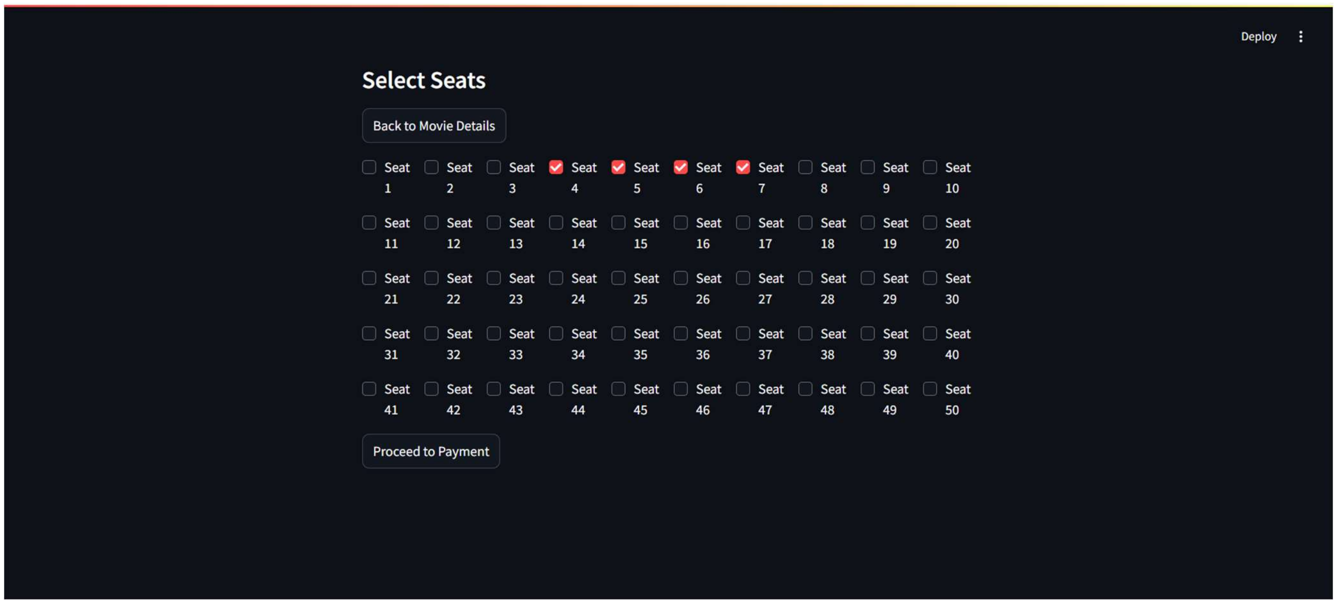
Director: Rajkumar Periasamy

Stars: Sivakartikeyan, Sai Pallavi

IMDb Rating: 8.6

Runtime: 169 mins

- Seat Selection :



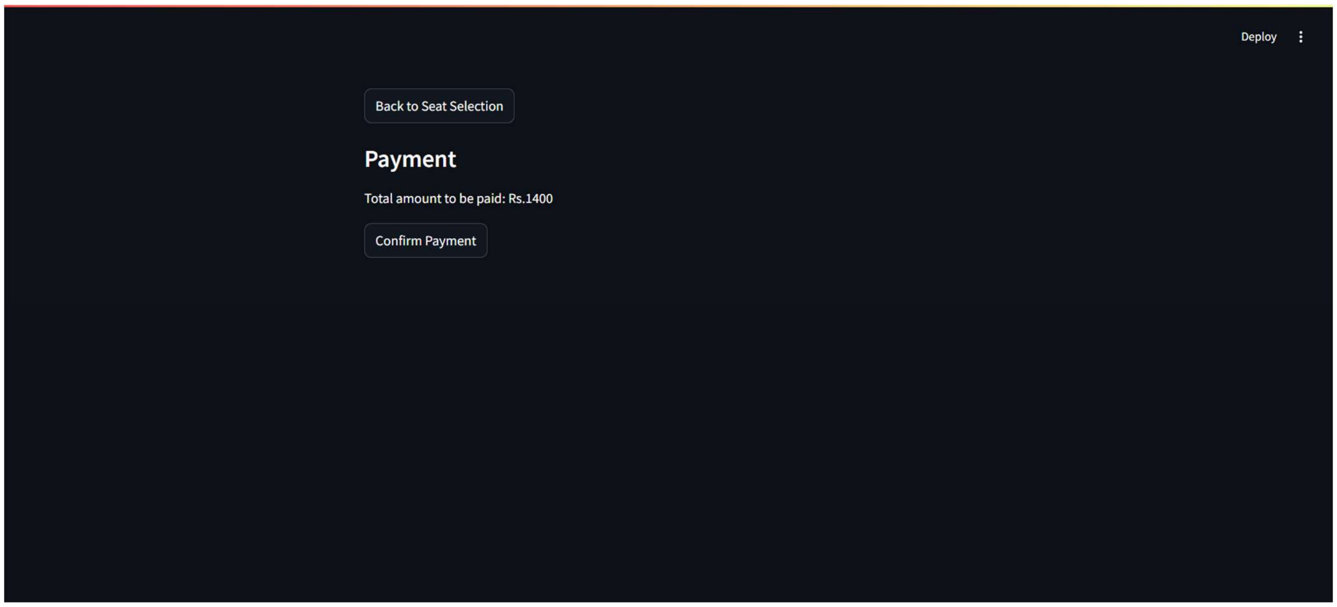
Select Seats

Back to Movie Details

☐ Seat 1	☐ Seat 2	☐ Seat 3	✓ Seat 4	✓ Seat 5	✓ Seat 6	✓ Seat 7	☐ Seat 8	☐ Seat 9	☐ Seat 10
☐ Seat 11	☐ Seat 12	☐ Seat 13	☐ Seat 14	☐ Seat 15	☐ Seat 16	☐ Seat 17	☐ Seat 18	☐ Seat 19	☐ Seat 20
☐ Seat 21	☐ Seat 22	☐ Seat 23	☐ Seat 24	☐ Seat 25	☐ Seat 26	☐ Seat 27	☐ Seat 28	☐ Seat 29	☐ Seat 30
☐ Seat 31	☐ Seat 32	☐ Seat 33	☐ Seat 34	☐ Seat 35	☐ Seat 36	☐ Seat 37	☐ Seat 38	☐ Seat 39	☐ Seat 40
☐ Seat 41	☐ Seat 42	☐ Seat 43	☐ Seat 44	☐ Seat 45	☐ Seat 46	☐ Seat 47	☐ Seat 48	☐ Seat 49	☐ Seat 50

Proceed to Payment

- Payment :



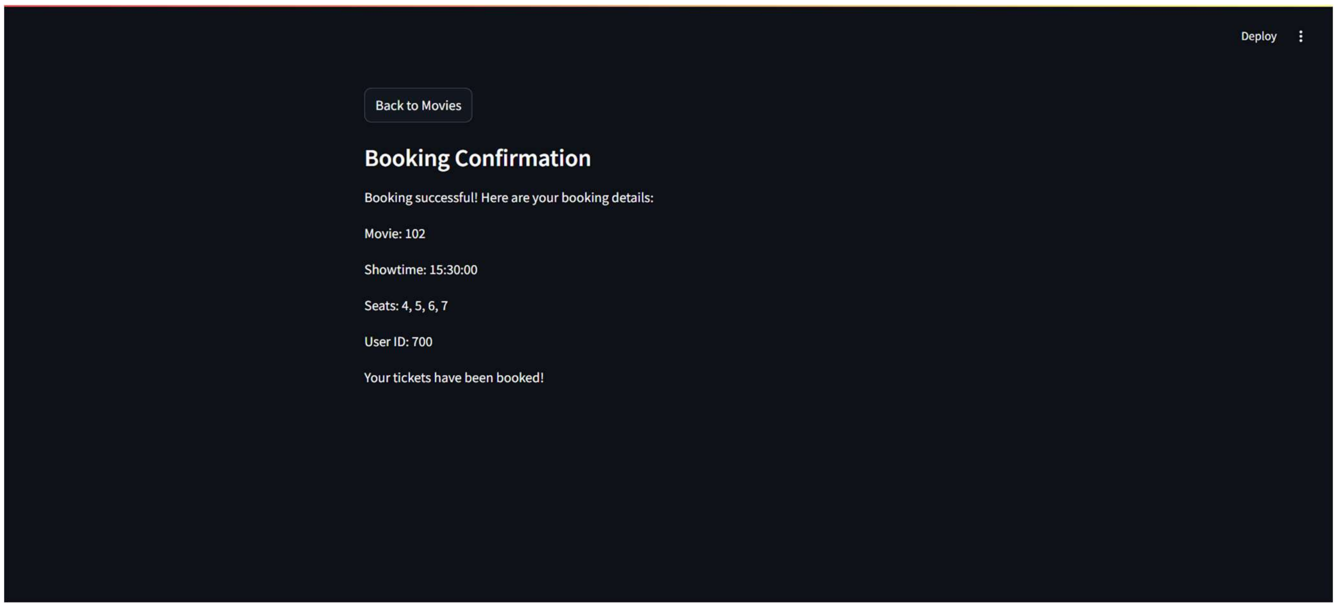
Back to Seat Selection

Payment

Total amount to be paid: Rs.1400

Confirm Payment

- Booking confirmation :



Booking Confirmation

Booking successful! Here are your booking details:

Movie: 102

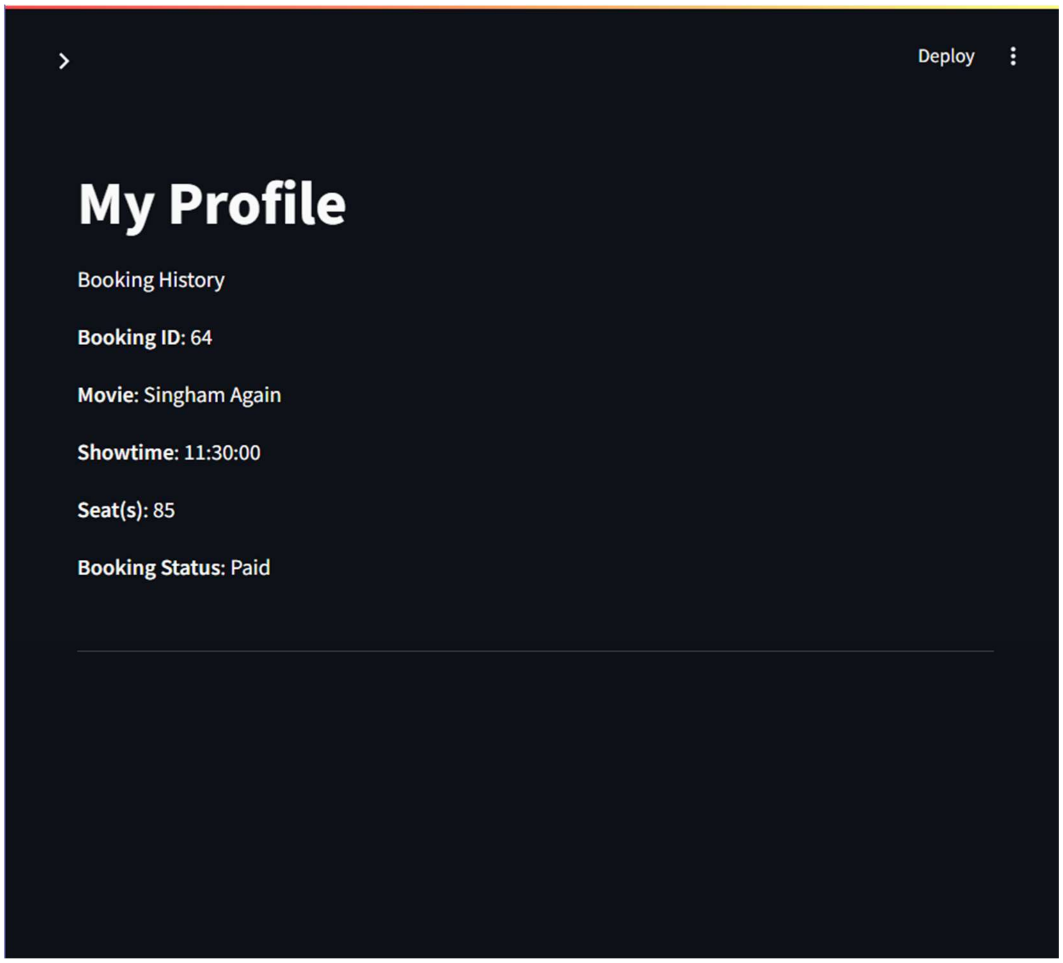
Showtime: 15:30:00

Seats: 4, 5, 6, 7

User ID: 700

Your tickets have been booked!

- Profile :



My Profile

Booking History

Booking ID: 64

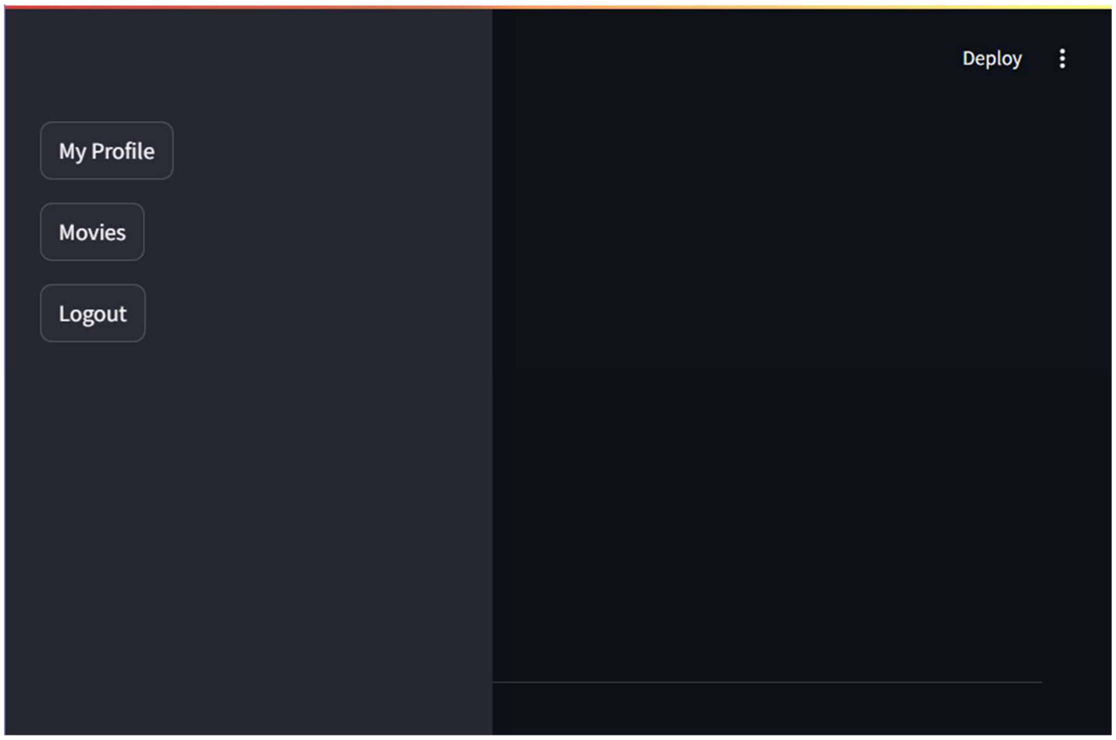
Movie: Singham Again

Showtime: 11:30:00

Seat(s): 85

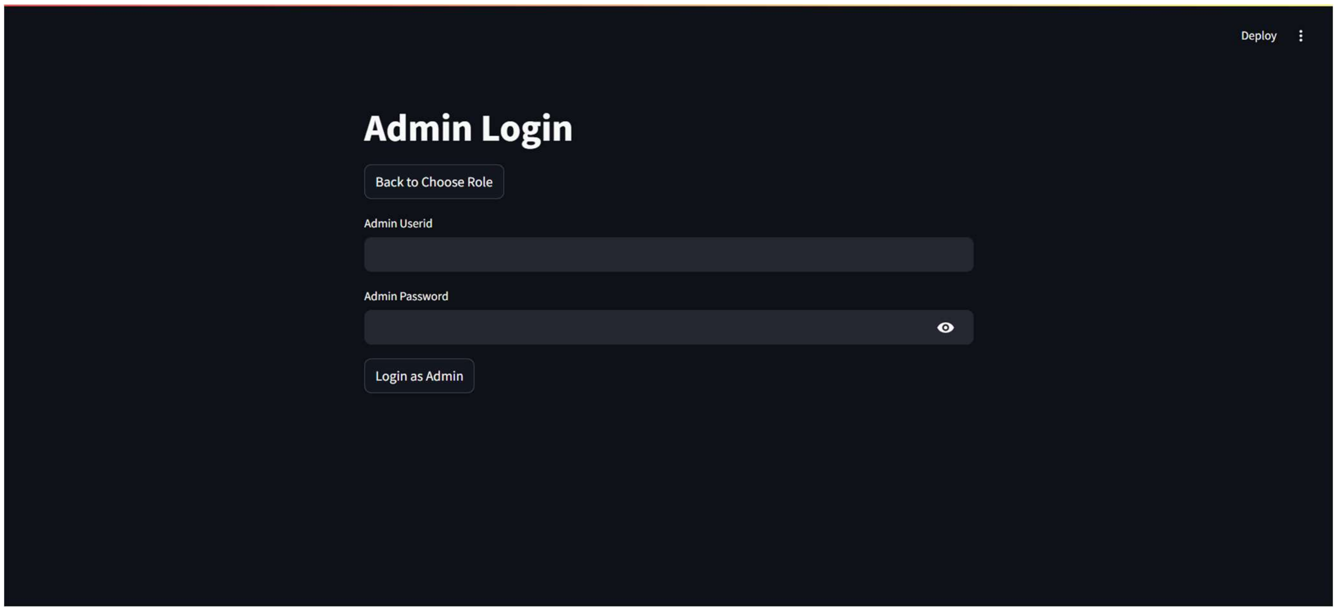
Booking Status: Paid

- Logout :

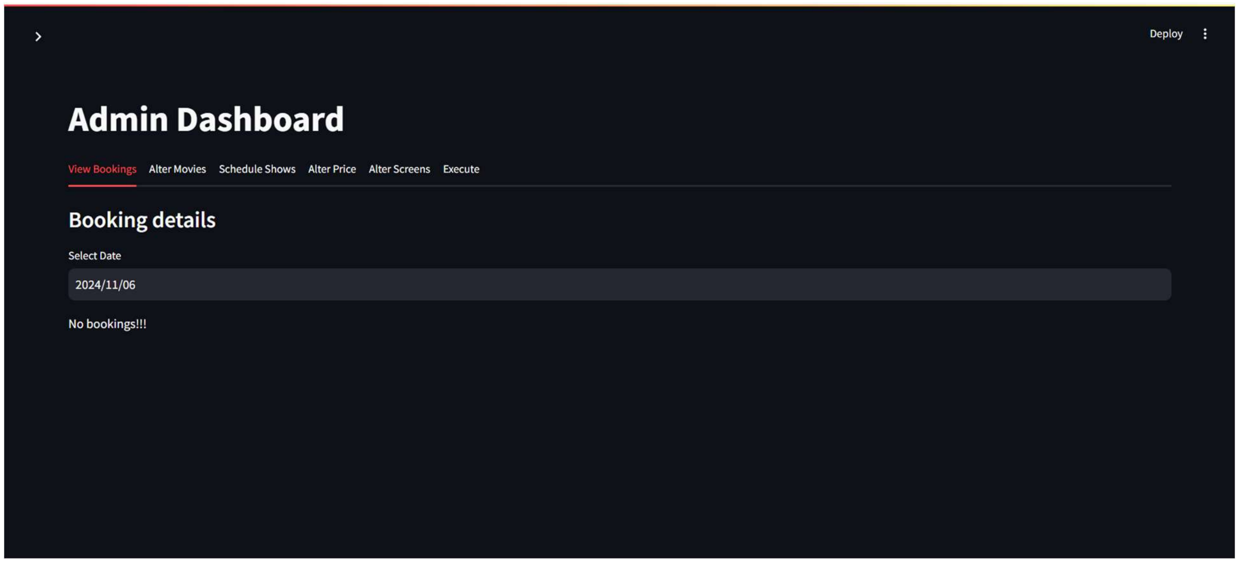


Admin Interface :

- Admin login :



- Admin dashboard :



Admin Dashboard

View Bookings | Alter Movies | Schedule Shows | Alter Price | Alter Screens | Execute

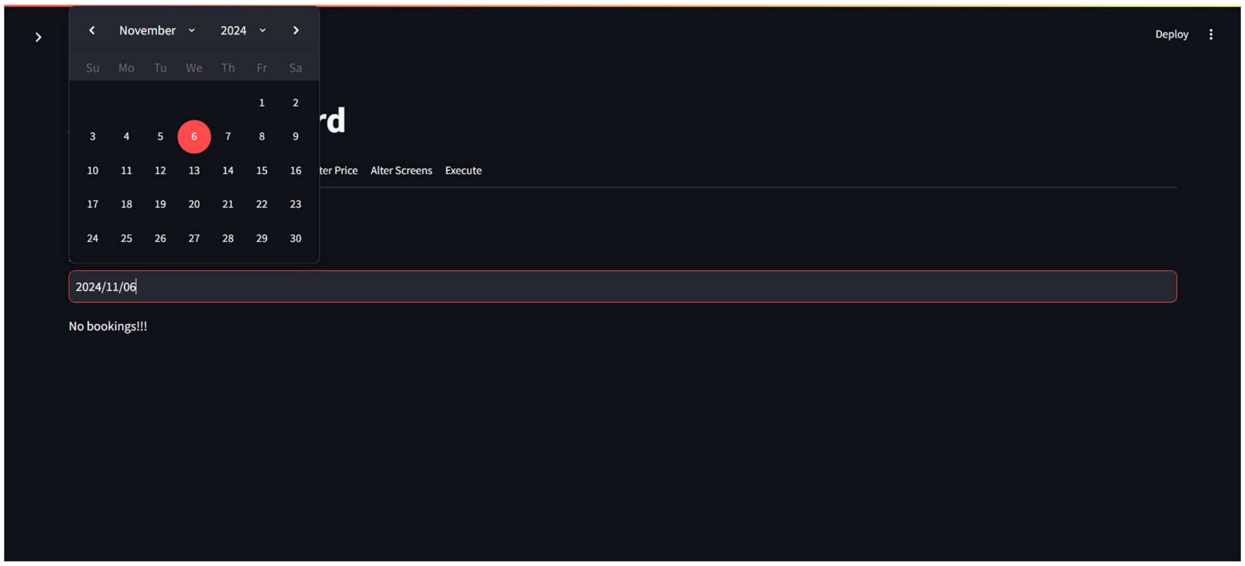
Booking details

Select Date

2024/11/06

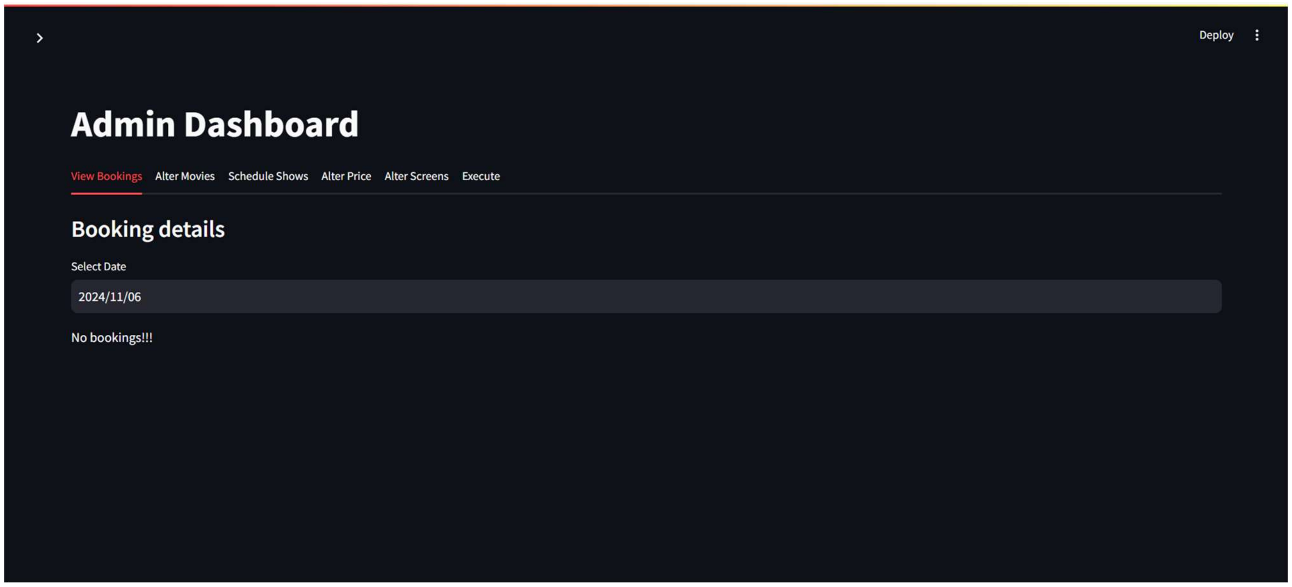
No bookings!!!

- View bookings :



2024/11/06

No bookings!!!



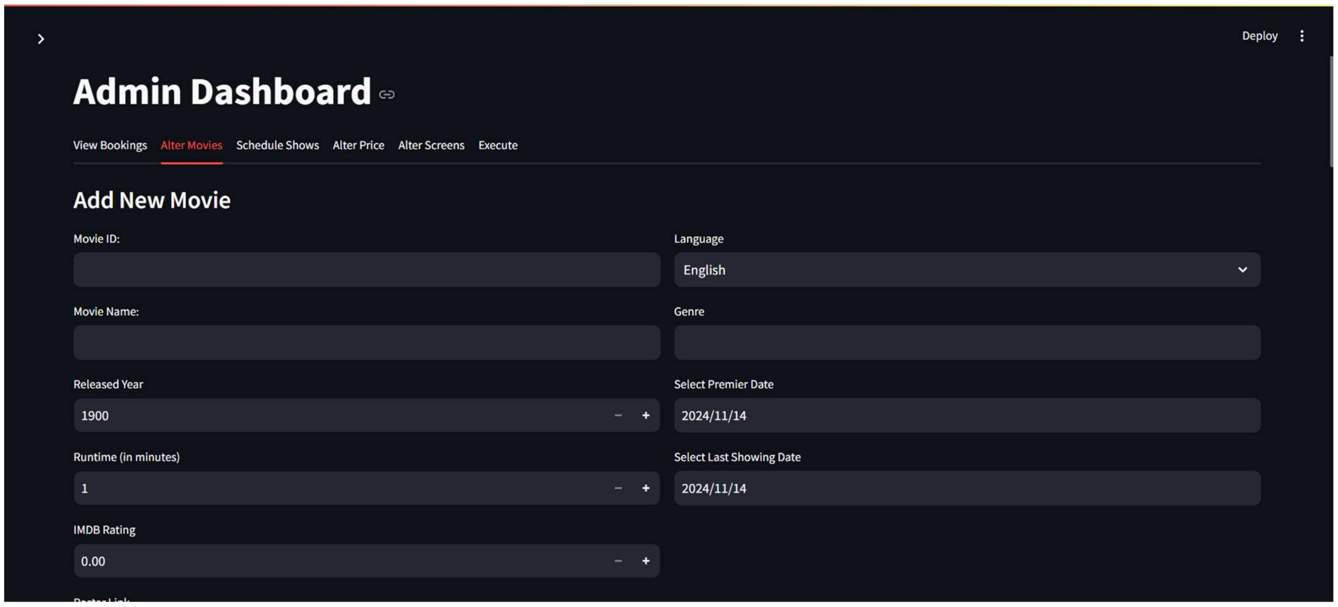
Admin Dashboard

Booking details

Select Date: 2024/11/06

No bookings!!!

- Alter movies :



Admin Dashboard

Add New Movie

Movie ID:

Language: English

Movie Name:

Genre

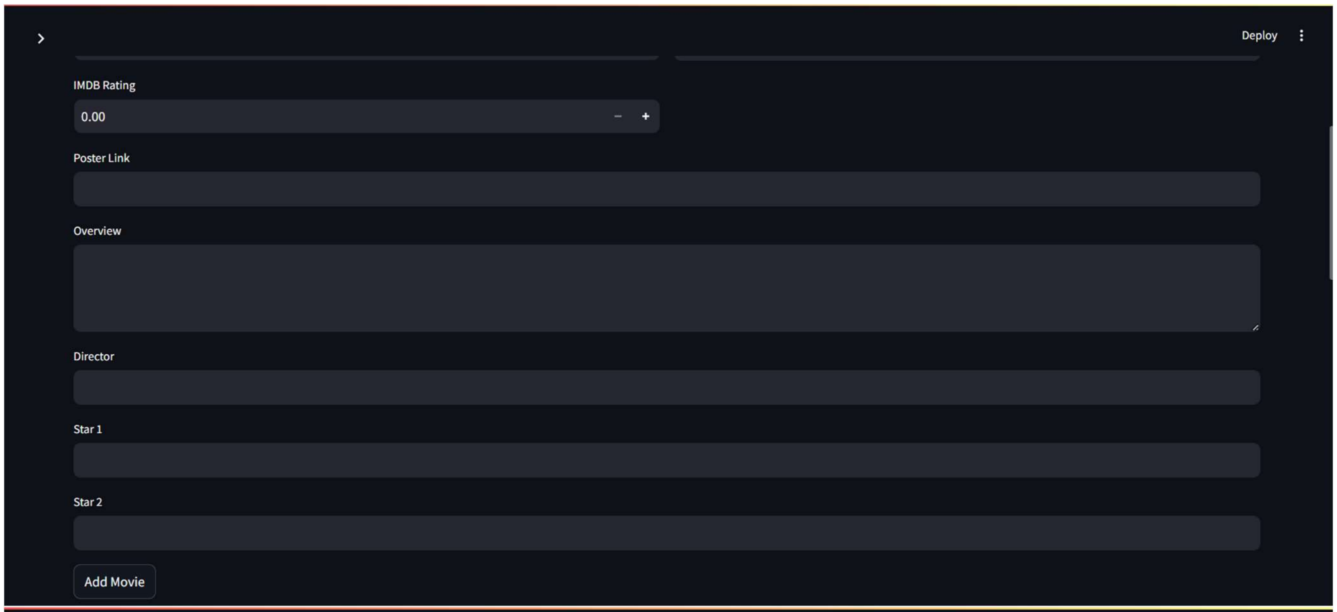
Released Year: 1900

Select Premier Date: 2024/11/14

Runtime (in minutes): 1

Select Last Showing Date: 2024/11/14

IMDB Rating: 0.00



IMDB Rating: 0.00

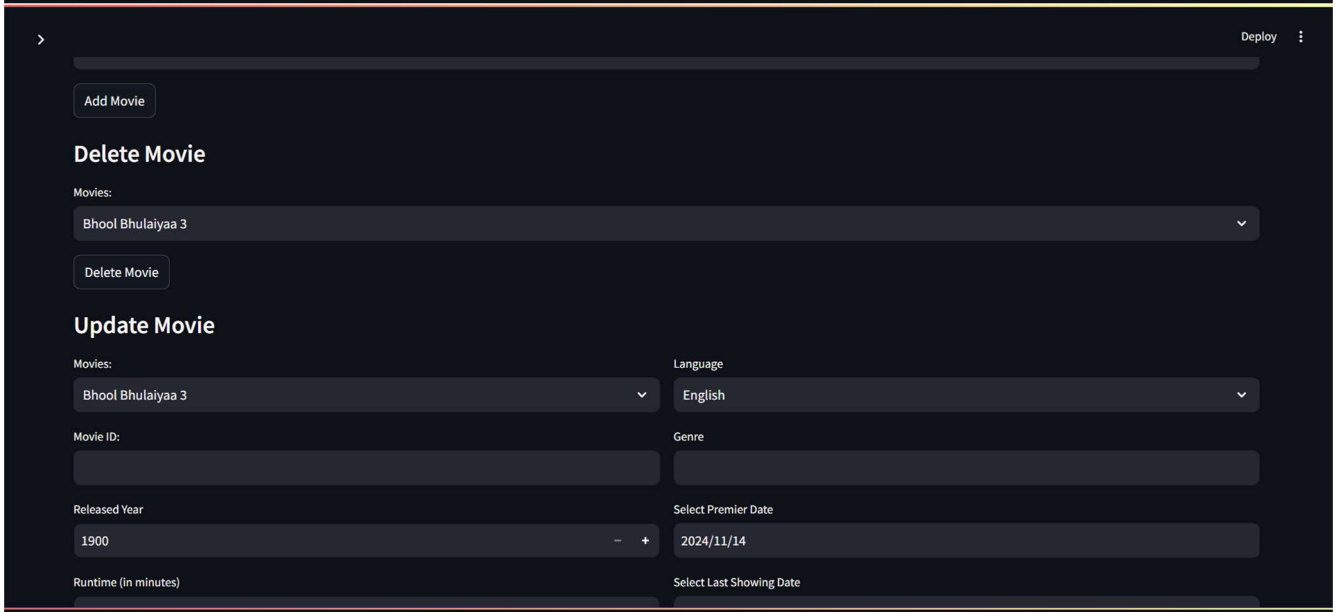
Poster Link

Overview

Director

Star 1

Star 2



Delete Movie

Movies: Bhool Bhulaiyaa 3

Update Movie

Movies: Bhool Bhulaiyaa 3

Language: English

Movie ID:

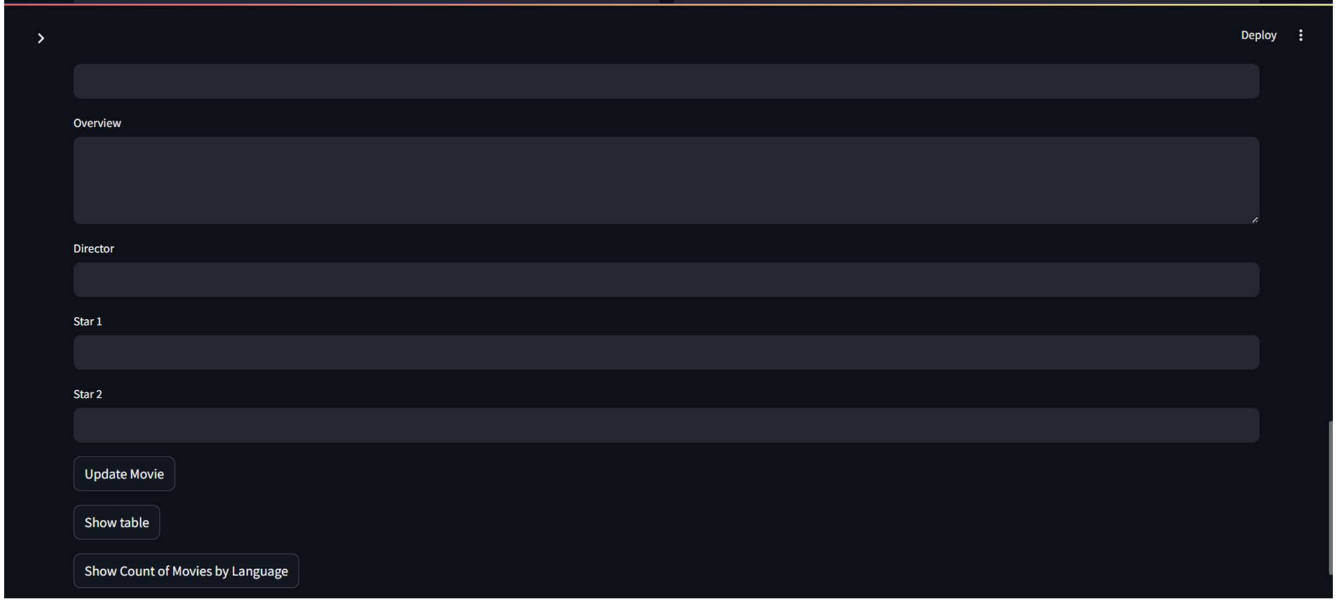
Genre

Released Year: 1900

Select Premier Date: 2024/11/14

Runtime (in minutes)

Select Last Showing Date



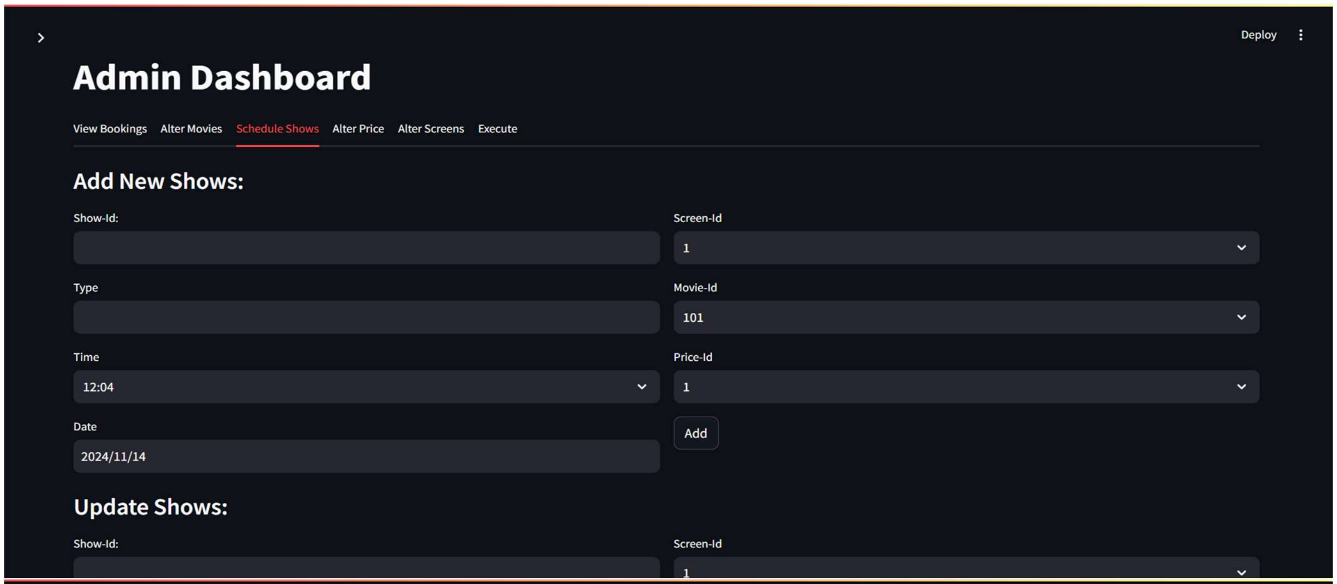
Overview

Director

Star 1

Star 2

- Schedule shows :



Admin Dashboard

Add New Shows:

Show-Id:

Screen-Id: 1

Type

Movie-Id: 101

Time: 12:04

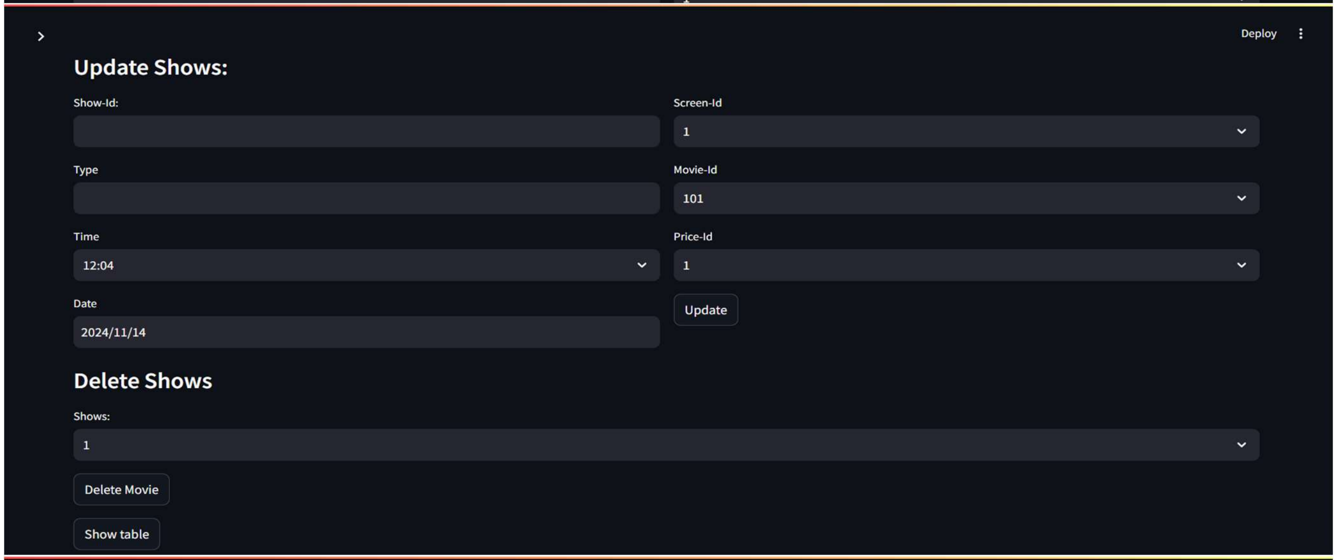
Price-Id: 1

Date: 2024/11/14

Update Shows:

Show-Id:

Screen-Id: 1



Update Shows:

Show-Id:

Screen-Id: 1

Type

Movie-Id: 101

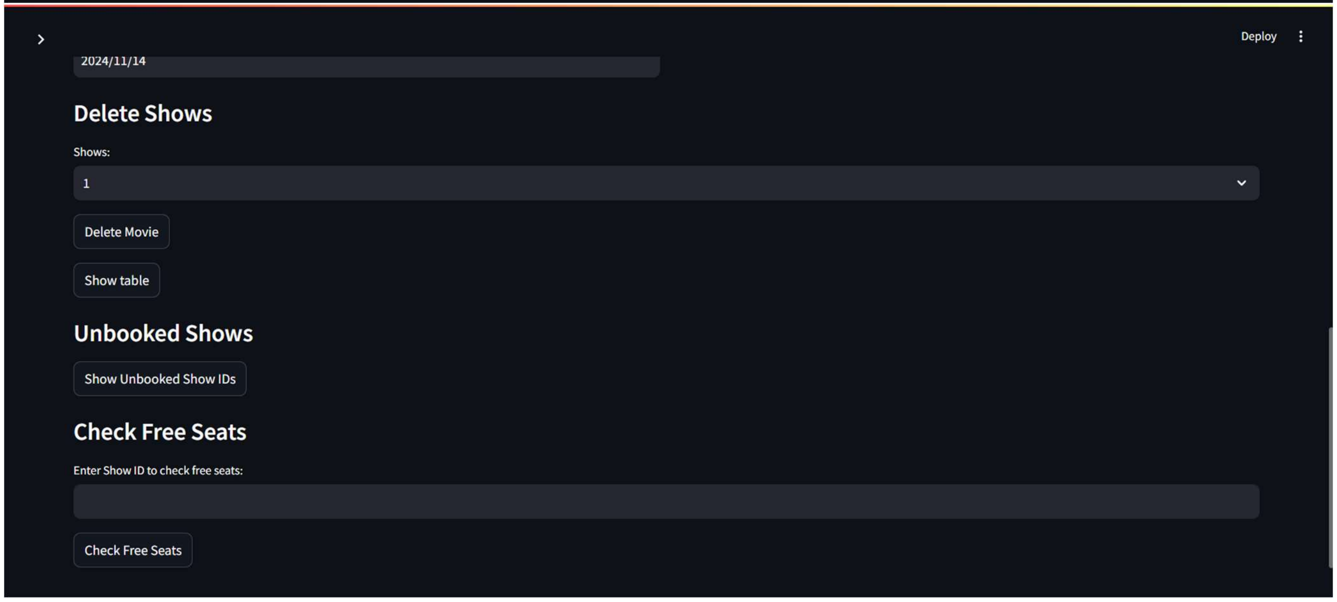
Time: 12:04

Price-Id: 1

Date: 2024/11/14

Delete Shows

Shows: 1



2024/11/14

Delete Shows

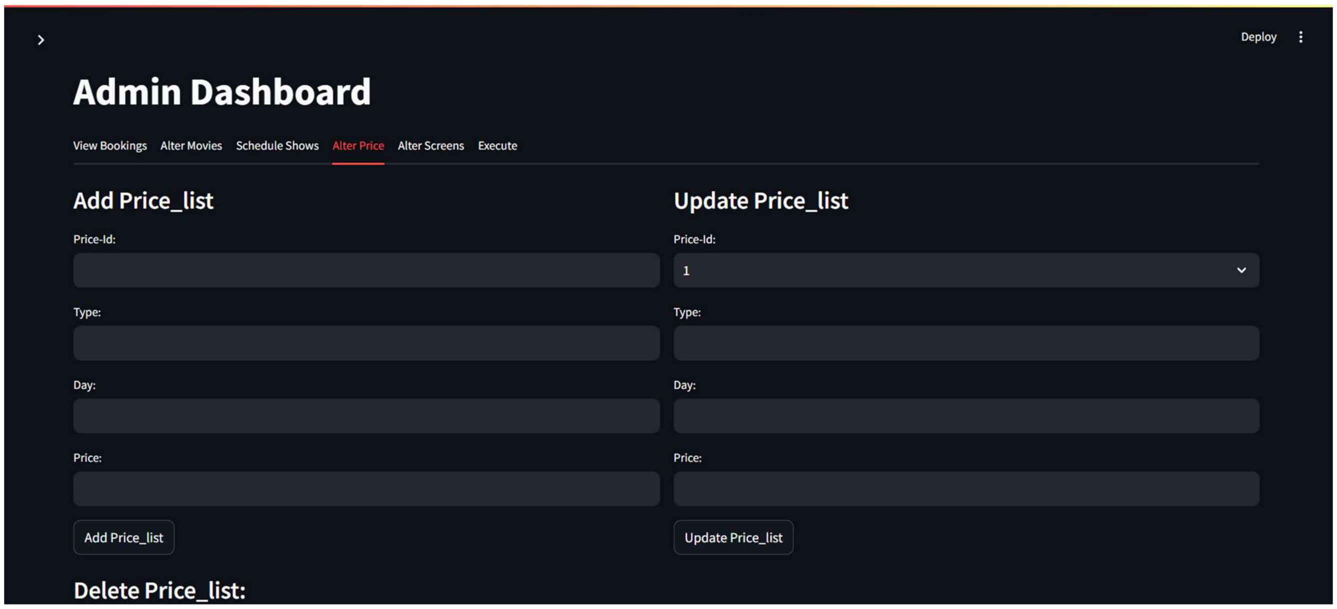
Shows: 1

Unbooked Shows

Check Free Seats

Enter Show ID to check free seats:

- Alter price :



Deploy

Admin Dashboard

View Bookings | Alter Movies | Schedule Shows | Alter Price | Alter Screens | Execute

Add Price_list

Price-Id:

Type:

Day:

Price:

Add Price_list

Update Price_list

Price-Id: 1

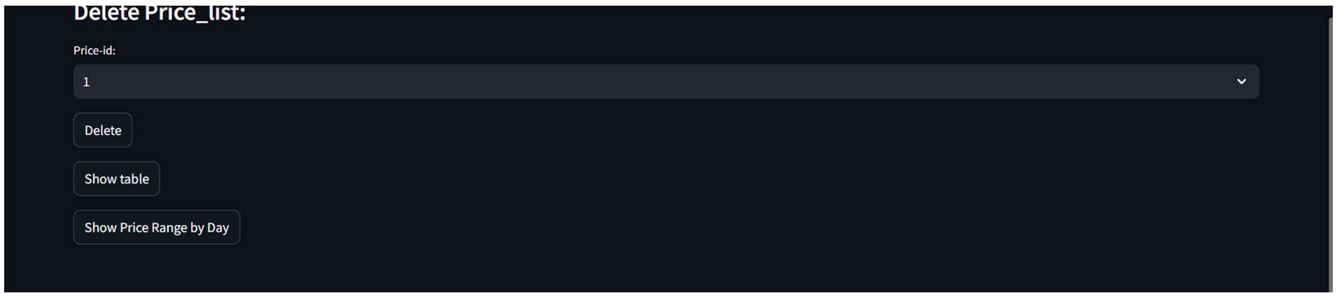
Type:

Day:

Price:

Update Price_list

Delete Price_list:



Delete Price_list:

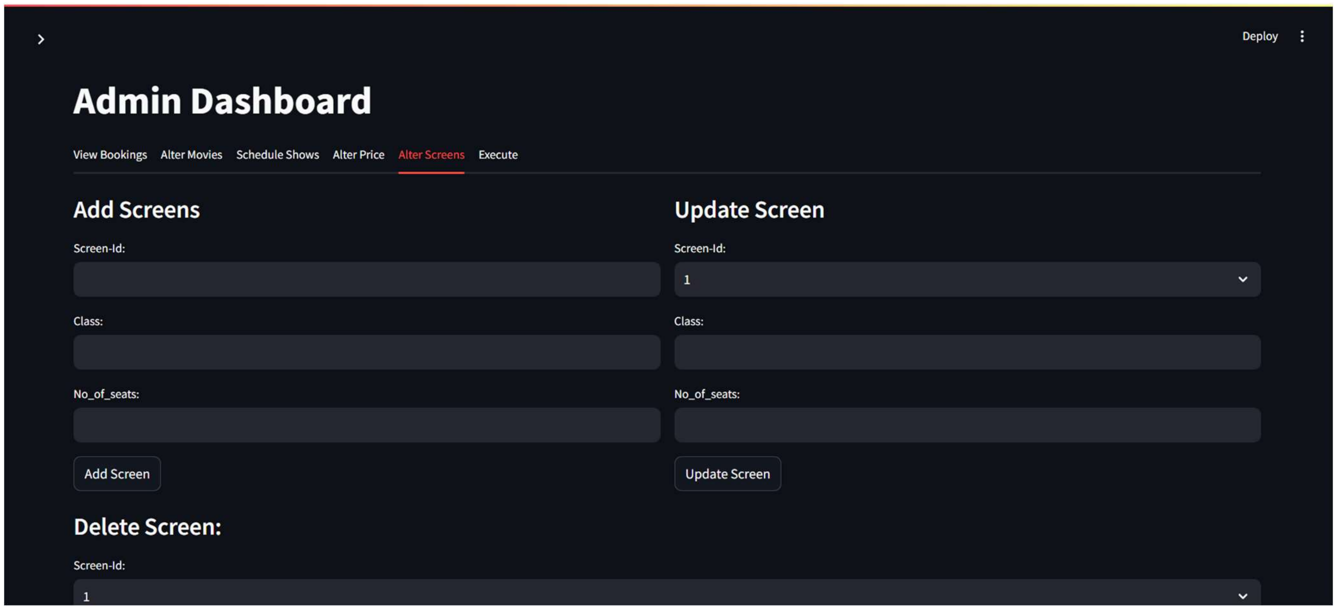
Price-id: 1

Delete

Show table

Show Price Range by Day

- Alter screen :



Deploy

Admin Dashboard

View Bookings | Alter Movies | Schedule Shows | Alter Price | Alter Screens | Execute

Add Screens

Screen-Id:

Class:

No_of_seats:

Add Screen

Update Screen

Screen-Id: 1

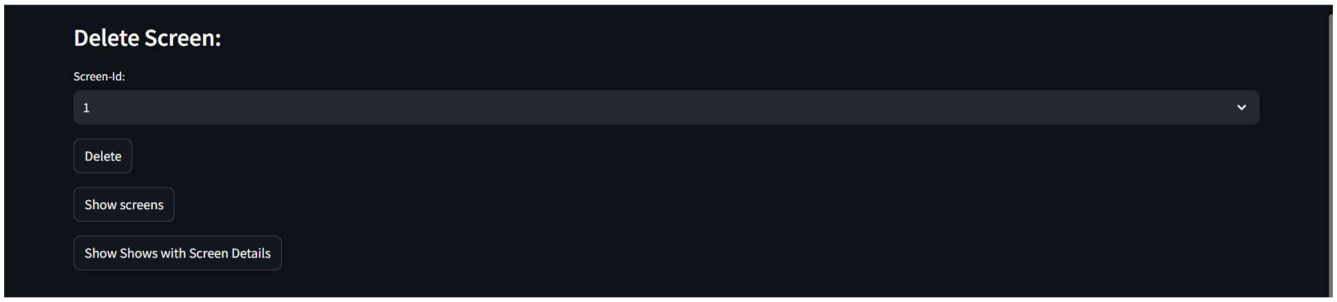
Class:

No_of_seats:

Update Screen

Delete Screen:

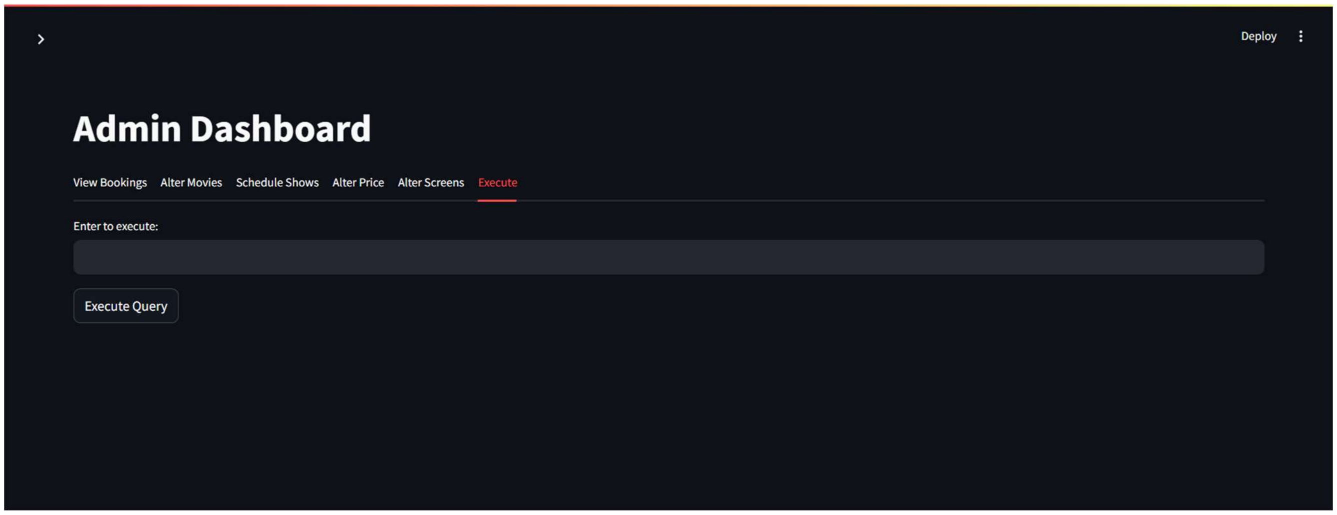
Screen-Id: 1


Delete Screen:

Screen-Id:

1

- Execute query :


Admin Dashboard

Enter to execute:

https://github.com/Raashi5013/dbms_project

***** Thank You *****